

Docket No.: 7675 (3225-034)



#22

PATENT

1053

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

RECEIVED
MAY 03 2002
Technology Center 2100

In re Application of

Keith BOLAND

Serial No. 09/086,627

Filed: May 29, 1998

Group Art Unit: 2153

Examiner: B. Edelman

For: METHOD AND APPARATUS FOR ALLOCATING NETWORK RESOURCES AND CHANGING THE
ALLOCATION BASED ON DYNAMIC WORKLOAD CHANGES

TRANSMITTAL OF APPEAL BRIEF

COMMISSIONER FOR PATENTS
Washington, DC 20231

Sir:

Submitted herewith in triplicate is Appellants' Appeal Brief in support of the Notice of Appeal filed
January 28, 2002. Please charge the Appeal Brief fee of \$320.00 to the attached credit card authorization form.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please
charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to
Deposit Account 07-1337 and please credit any excess fees to such deposit account.

Respectfully submitted,

LOWE HAUPTMAN GILMAN & BERNER, LLP

Kenneth M. Berner
Registration No. 37,093

USPTO Customer No. 22429
1700 Diagonal Road, Suite 310
Alexandria, Virginia 22314
(703) 684-1111
(703) 518-5499 Facsimile
Date: April 29, 2002
KMB/lcw

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Keith BOLAND

Serial No. 09/086,627

Filed: May 29, 1998

For: METHOD AND APPARATUS FOR ALLOCATING NETWORK RESOURCES
AND CHANGING THE ALLOCATION BASED ON DYNAMIC WORKLOAD
CHANGES

Box Patent Appeal Brief
Commissioner for Patents
U.S. Patents and Trademarks Office
Washington D.C. 20231
Attn: BOARD OF PATENT APPEALS AND INTERFERENCES

APPELLANT'S BRIEF (37 C.F.R. § 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on January 28, 2002.

The fees required under § 1.17(f) and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate.

This brief contains these items under the following headings, and in the order set forth below (37 C.F.R. § 1.192(c)):

I. Real Party in Interest.

II. Related Appeals and Interferences.

III. Status of Claims.

IV. Status of Amendments.

V. Summary of Invention.

VI. Issues.

 A. First Issue

 B. Second Issue

VII. Grouping of Claims.

VIII. Arguments.

 A. First Issue 1. *35 U.S.C. 103(a)* rejection of claims 1-5 and 9-16 as being unpatentable over Culbert in view of Sumimoto
 2. Conclusion

 B. Second Issue 1. *35 U.S.C. 103(a)* rejection of claim 7 as being unpatentable over Culbert in view of Sumimoto and further in view of Hauser
 2. Conclusion

IX. Appendix of Claims Involved in the Appeal.

The final page of this brief bears the attorney's signature.

I. REAL PARTY IN INTEREST

The real party in interest in this appeal is NCR Corporation of 101 W. Schantz Avenue, Dayton, OH 45479-0001.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There is a total of 14 claims in the application, which are identified as claims 1-5, 7 and 9-16.

B. Status of all the claims

1. Claims cancelled: claims 6 and 8
2. Claims withdrawn from consideration but not cancelled: none
3. Claims pending: claims 1-5, 7 and 9-16
4. Claims allowed: none
5. Claims rejected: claims 1-5, 7 and 9-16

C. Claims on Appeal

Claims on appeal are claims 1-5, 7 and 9-16 as rejected by the Final Office Action dated September 27, 2001.

IV. STATUS OF AMENDMENTS

Appellant filed a Request for Reconsideration on December 27, 2001 in response to the September 27, 2001 Final Office Action. The Request for Reconsideration did not include claim amendments.

V. SUMMARY OF INVENTION

The invention relates to a method and apparatus for allocating network resources and for changing the resource allocation based on dynamic workload changes. See page 1, lines 7-10 of the specification.

The present invention stems from a realization that conventional methods of allocating resources on a computer network do not ensure that a particular process, i.e. a high-priority process, will have a minimum amount of required network resources. See page 2, lines 10-11 of the specification. Critical situations occur when one or more nodes on the network fail(s) and a network server redistributes the processing load among the surviving nodes. If one of the conventional methods of allocating network resources is implemented on the network, a high-priority process or group of processes may not have sufficient network resources allocated to either run on the network, or run efficiently, while a lower-priority process may have network resources automatically reallocated thereto which should more preferably be reallocated to the high-priority process or group of processes. See page 2, lines 15-23 of the specification.

Thus, it is a feature and advantage of the present invention to allocate a minimum amount of network resources to a process or group of processes irrespective of the requirements of any other process or group of processes running on the computer network, preferably, based on prioritization of all processes running on the computer network. See page 3, lines 6-8 and 12-15 of the specification. The present invention is suitable for allocating and switching network resources based on dynamic workload changes, especially

when network resources used by a high-priority process or group of processes become unavailable.

The method of allocation network resources of the present invention can be implemented in network 100 (Fig. 1) which has a plurality of nodes e.g. 104, 106 through 114. Each node is a computer system having a general configuration as shown in Fig. 2. A local resource manager 104a-114a (Fig. 3) resides in e.g. processor 204 (Fig. 2) of each of the respective nodes 104-114 to monitor the operational states thereof. See page 9, lines 26-28 of the specification. Local resource managers 104a-114a of the respective nodes 104-114 communicate with domain resource manager 106c (Fig. 3). See page 10, lines 15-18 of the specification. Resource-related data, including but not limited to hardware information, reliability information, processor resources/usage, memory resources, storage resources, printer resources and the like, is transmitted from local resource managers 104a-114a to domain resource manager 106c. See page 10, lines 18-26 of the specification. Network 100 further includes scheduler 280 (Figs. 3-4) which can reside in a node e.g. 104 (Fig. 3) for the distribution of a plurality of processes 320a-320d (Fig. 5) on network 100. See page 11, lines 8-9 of the specification.

In accordance with the method of the present invention, a priority is assigned to each of processes 320a-320d so that e.g. process 320a has first priority, process 320b has second priority and so forth. The process priorities are stored in element 280f (Fig. 4) of the scheduler 280. See page 13, lines 2-4 of the specification. A minimum resource allocation is set for each of processes 320a-320d. For example, process 320a is allocated at least sixty percent of processor 204 on node 110 and one hundred percent of processor 204 on node 112. See Fig. 5. Importantly, the minimum resource allocation for first priority process 320a on the at least two nodes is set independent of the computer resources needed by other processes i.e. 320b-320d running on the network. The minimum resource allocations for processes 320a-320d are stored in element 280c of scheduler 280. See page 12, lines 26-28 of the specification. Operational states of nodes 104-114 are closely monitored by local resource managers 104a-114a and domain resource manager 106c which work in conjunction with scheduler 280 to reallocate network resources. See page 13, lines 19-21 of the specification.

If, for example, node 110 fails, there will no longer be sufficient network processing resources to meet the needs of processes 320a-320d. Because process 320a has the highest priority, the minimum resource allocation for process 320a which is stored in element 280c must be **guaranteed**. In the simple example of Fig. 5, process 320a will be moved to either of nodes 108 and 114 because process 320d running thereon has the lowest priority. After process 320a is moved, the amount of network resources actually allocated to process 320d will decrease accordingly. *See* page 13, lines 21-26 of the specification and claims 1, 11-13. In other words, network resources will be redistributed **irrespective of** an amount of computer resources necessary for lower-priority process 320d to run on the computer network. *See* page 3, lines 12-15 of the specification and claims 14 and 16.

VI. ISSUES

A. First Issue

Whether the *35 U.S.C. 103(a)* obviousness rejection of claims 1-5 and 9-16 over Culbert (U.S. Patent No. 5,838,968) in view of Sumimoto (U.S. Patent No. 5,522,070) should be reversed ?

B. Second Issue

Whether the *35 U.S.C. 103(a)* obviousness rejection of claim 7 over Culbert in view of Sumimoto and further in view of Hauser (U.S. Patent No. 5,889,956) should be reversed ?

VII. GROUPING OF CLAIMS

For purposes of this Appeal Brief only, the claims have been grouped as follows:

First Issue:

Group I. Claims 1-5, 9-13, 15

Group II. Claim 14

Group III. Claim 16

Second Issue:

Group IV. Claim 7

The Appellant respectfully asserts that claims in each group are separately patentable, and thus, the claims do not stand or fall together.

VIII. ARGUMENTS

A. First Issue

35 U.S.C. 103(a) rejection of claims 1-5 and 9-16 as being unpatentable over Culbert in view of Sumimoto

The Examiner rejected claims 1-5 and 9-16 under *35 U.S.C. 103(a)* as being unpatentable over Culbert in view of Sumimoto. Appellant respectfully traverses this erroneous rejection because the references, singly or in combination, fail to teach or disclose all limitations of the rejected claims.

It is important to emphasize that, in accordance with the present invention, should insufficient network resources be available, the minimum resource allocation for the first (high-priority) process must be **guaranteed** (as recited in claim 1) **irrespective of** an amount

of computer resources necessary for the second (low-priority) process to run on the computer network (as recited in claims 14 and 16).

The Culbert method, especially as described in col. 9, lines 15-54, col. 10, line 48 through col. 11, line 6, and col. 11, lines 46-52, can be summarized as follows:

(a) The primary object of the Culbert resource allocation system is to assure that allocation of resources, and hence system performance, is globally optimized and dynamically managed. See col. 3, lines 23-31.

(b) Each task, e.g. 350 (Fig. 3) is associated with a task resource utilization vector 300 (Fig. 3) which contains multiple task resource utilization records 310, 320, 330 (Fig. 3) sorted in preferred order. The first record, 310, is the most favorable allocation of resources that task 350 could use. The last record 330 specifies the minimal resource utilization configuration below which the task 350 will fail due to resource constraints. See col. 7, lines 40-44, col. 7, line 66 through col. 8, line 9, and col. 8, lines 30-32.

(c) If the system becomes resource constrained, and tasks have difficulty gaining access to needed resources, a resource manager must decide whether to lower the available resources for current tasks, or fail a resource allocation request. In the former instance, resource degradation occurs. See col. 9, lines 15-23. Tasks are degraded as specified in col. 9 lines 27-35. In particular, tasks with lower priority will always be degraded as much as possible before any high priority task. It has also been noted that the task prioritization in Culbert is used solely to determine the order in which tasks are degraded, and not for any other purposes.

(d) When resource degradation is necessary, i.e. no resources are available to fulfill a resource allocation request, each task is asked what resources it would provide to the system if were asked to. The task being queried may respond with one of several answers as specified in col. 10, line 59 through col. 11, line 6. In particular, the task being queried can respond that it cannot be changed and cannot give up any resources.

(e) In the worst scenario, when the queried tasks deny to give up their resources or if the amount of recovered resources is not sufficient, the resource allocation request will be denied. See col. 11, lines 51-52.

The subsequent notes are believed to logically follow from the disclosure of Culbert.

(f) When a resource deficit, e.g. due to a hardware failure, occurs in the Culbert system, there will be two types or groups of tasks which may be affected by the resource deficit i.e. those which directly suffer from the resource deficit, e.g. because they have lost resources due to the hardware failure, and those which do not directly suffer from the resource deficit. The tasks in the first group will then ask the system to allocate additional resources thereto. The tasks in the second group may indirectly suffer from the resource deficit when they have to give up part of their current resources to the first group tasks. See points (c) and (d).

(g) However, the second group tasks in the method of Culbert would not give up more than they could afford. Culbert does not teach forcibly taking resources away from the second group tasks even if they have a priority lower than that of the first group tasks. All Culbert teaches is to ask the second group tasks to give up some of their resources and move to a lower run level, if they could. See point (d). Culbert is also silent on the possibility of voluntary termination of a task even if it is a low-priority task. All of the above lead to a logical conclusion that each second group task will not be terminated or self-terminated as a result of the resource deficit, and should be left, at least, with an amount of resources necessary for its lowest run level. See point (b). That is, a minimum resource allocation is guaranteed for each second group task, regardless of what priority level the task has.

(h) However, a minimum resource allocation is not guaranteed for a first group task even if it has a high priority level. In the very worst scenario when the amount of computer resources allocated to a high-priority first group task drops below a minimum required level and the high-priority first group task sends a request for additional resources, the request could still be denied. See point (e). This situation may happen even if the remaining amount of resources in the system is sufficient to satisfy the request for additional resources of the high-priority first group task, because the remaining resources are occupied by other, possibly, lower-priority tasks which do not want or cannot give the remaining resources up. That is, in accordance with the Culbert management system, a minimum resource allocation necessary for a high-priority process to run properly might not be guaranteed.

(i) In other words, in the global performance optimizing method of Culbert, assurance of a minimum resource allocation for a task is based not on a priority level of the task, as presently claimed, but on the status of the task at the time the resource deficit occurs i.e. the minimum resource allocation is guaranteed for the task if it is a second group (resource-providing) task and is not guaranteed if the task is a first group (resource-requesting) task. Such a method is clearly different from what described and claimed in the instant application.

(j) Moreover, the facts that (1) the Culbert system has to ask the second group tasks for resources and (2) the asked second group tasks may refuse to give up any resources and the system could do nothing about it, see point (d), mean that Culbert does not disregard minimum resource allocations for the second group tasks even if they are at priority levels lower than those of the first group tasks. In other words, the minimum resource allocations of the second group tasks must be taken into account during a degradation process after the resource deficit occurs.

Appellant will now address the Examiner's rejections in turn.

With respect to claim 1, the Examiner in pages 2-3 of the Final Office Action alleged that the Culbert method, as summarized above, includes all limitations of claim 1. The Examiner, however, admitted that the Culbert method, unlike the claimed invention, is being implemented in a single computer system rather than being arranged to manage resources across a network including at least two nodes. The Examiner also noted that Sumimoto discloses a system for managing resource allocation for processes distributed across a network or multiple computers. The Examiner then concluded that "it would have been obvious to use the resource allocation scheme taught by Culbert for processes and resources distributed throughout a network, as taught by Sumimoto, so that the most important network distributed processes can be assured available resources". See page 3, third paragraph of the Final Office Action. In the Advisory Action dated January 8, 2002, the Examiner further clarified that his suggestion or motivation to combine/modify the references is found in Culbert, especially col. 7, lines 49-51, col. 9, lines 52-54 and 24-32. See the paragraph bridging pages 2-3 of the Advisory Action.

Without analyzing the Examiner's questionable suggestion/motivation in detail, Appellant respectfully submits that the applied patents, especially Culbert, fail to disclose, teach or suggest all limitations of claim 1, especially the feature that the minimum resource allocation for the first (higher-priority) process is **guaranteed** should insufficient network resources be available.

The present invention relates to situations when network resources are in sufficient, and thus, re-allocation of network resources will be required. If this is the case, the minimum resource allocation for the first (high-priority) process must be guaranteed. See claim 1 and the working example illustrated in Fig. 5. In contrast, Culbert relates to optimization of global system performance where a request for additional resources could be denied if this serves the goal of maintaining the global optimization of the whole computer system. See point (e). In the very worst scenario discussed at point (h) above, it is clear that, in the Culbert management system, the minimum resource allocation for the high-priority process to run properly is **not guaranteed**.

The Examiner did not agree with Appellant's reading of the Culbert reference. More particularly, the Examiner noted that Culbert, in col. 9, lines 47-54, teaches optimizing the system performance while still providing the needed resources, and concluded that a minimum resource allocation is indeed guaranteed. See page 8, the first full paragraph of the Final Office Action. The Examiner seemed to take col. 9, lines 47-54 out of the context of the Culbert reference. When read in light of the whole description of the Culbert method, especially in col. 9 lines 24-54, the relevant teaching relied upon by the Examiner means that minimum resource allocations are guaranteed only for those processes which do not require additional resources when a resource deficit occurs. For processes which, for example, have lost resources and sent requests for additional resources, their requests for additional resources could still be denied, and hence, their minimum resource allocations could not be guaranteed, as exemplified by the above mentioned very worst scenario. See also point (i). (This argument was presented in the December 27, 2001 Amendment, and has not been responded to by the Examiner.)

It should now be clear that the Culbert resource management system does not guarantee minimum resource allocations for high-priority processes in the manner claimed in

independent claim 1. The missing element is neither supplied by other applied and/or cited references of record. Accordingly, Appellant respectfully submits that claim 1 is patentable over the applied art of record, and requests that the erroneous rejection of claim 1 under 35 U.S.C. 103(a) be withdrawn.

Claims 2-5, 7, and 9-16 are considered patentable at least for the reason advanced with respect to independent claim 1. Claims 2-5, 7, and 9-16 are also patentable on their own merits since these claims recite other features of the invention neither disclosed, taught nor suggested by the applied art.

With respect to claim 14, the applied references, especially Culbert, fail to disclose, teach or suggest the claimed redistributing step in accordance with which an amount of computer resources previously assigned to the second (lower-priority) process is removed and reallocated to the first (high-priority) process **irrespective of** an amount of computer resources **necessary for the second process to run** on the computer network. See claim 14. It is acknowledged that Culbert teaches (voluntary) removal of computer resources from second group tasks to resource-hungry first group tasks. However, this is implemented not without taking into account the minimum resource allocations of the second group tasks. See point (j).

The Examiner agreed with Appellant's position that the Culbert degradation process is not done wholly irrespective of an amount of computer resources necessary for the second (low-priority) process to run on the computer network. The Examiner, however, stated that doing so is a mere design choice. See page 9, lines 2-6 of the Final Office Action.

Appellant in the December 27, 2001 Amendment has traversed the Examiner's "design choice" argument, and requested that citation of a particular reference or references which might support the Examiner's assertion be provided. Up to now, Appellant's request has not been fulfilled. Accordingly, Appellant must conclude that the Examiner's "design choice" argument is merely a conclusory and unsupported statement which cannot be relied upon in a determination that the claimed invention of claim 14 is an obvious variant of the hypothetical Culbert/Sumimoto combined method. Moreover, Appellant has also noted that none of the applied/cited references in the instant application supply the desirability of reallocating computer

resources in a network irrespective of resources necessary for low-priority processes to run, as presently claimed. Without the benefit of the disclosure of the present invention, a skilled artisan would not have been motivated to modify the Culbert system to reallocate computer resources in the claimed manner.

Appellant in the December 27, 2001 Amendment has also pointed out that it is impermissible to modify the Culbert system to reallocate resources wholly irrespective of an amount of computer resources necessary for a low-priority process to run on the computer network, because such a modification runs counter to the requirements of the Culbert system itself. See also point (j). In other words, modifying Culbert to function in the presently claimed manner would impermissibly change the principle of operation of the prior art reference being modified, i.e. Culbert.

The Examiner did not agree with Appellant's latter argument. In particular, the January 8, 2002 Advisory Action, in the paragraph bridging pages 4-5, stated that

1. Primarily, the Culbert system can be modified in any way that might be desirable to a designer of the system.
2. Culbert has opted to allow low-priority processes to always maintain some amount of allocated resources for proper performance. However, Culbert could have just as easily designed the system to allow low-priority processes to be completely overridden by high-priority processes.
3. For example, the priority of the low-priority tasks could have been set to zero, or the minimum resources necessary could have been set to zero.
4. Either way, the choice to completely override low-priority processes instead of protecting some minimum resource allocation for these low-priority tasks is not an inventive step but simply a choice of design regarding a desired level of resource allocation.
5. Therefore, it would be permissible, and even desirable, to reallocate resources in the Culbert system wholly irrespective of an amount of computer resources necessary for low-priority processes to run, if the only important processes for which a designer desires to allocate resources are high-priority processes.

Regarding points 1 and 4, it should be again noted that the Culbert system could be modified only if there is a suggestion or motivation to do so. Currently, there is no evidence in the records that such a suggestion or motivation might be found in the references themselves or in the knowledge generally available in the art.

Regarding point 2, the Examiner's statement that Culbert could have easily designed his system otherwise is unsupported. The reference, in fact, unequivocally teaches away from being modified in the presently claimed manner. See point (j) and claim 14.

Regarding point 3, the Examiner's proposal of possible modifications of the Culbert system would work only if there were suggestion or motivations to do so. Moreover, it is not clear how setting of the priority of low-priority tasks to zero is related to the claimed invention. Setting of the minimum resource allocations for low-priority tasks to zero is not contemplated by Culbert. It should be noted that both Culbert and the present invention take into consideration the (minimum) amount of computer resources **necessary for a (low-priority) process to run**. See point (b) and claim 14. A zero amount of resources cannot be considered as necessary for a process, even a low-priority one, to run.

Regarding point 5, it should be again noted that Culbert is not only interested in high-priority processes. Instead, the reference is concerned with global optimization of the whole system. See point (a).

For the overwhelming reasons advanced above, Appellant respectfully submits that claim 14 is patentable over the applied art of record, and requests that the erroneous rejection of claim 14 under *35 U.S.C. 103(a)* be withdrawn.

With respect to claim 16, the applied references, especially Culbert, fail to disclose, teach or suggest the claimed method in accordance with which the minimum resource allocation for the high-priority process is **guaranteed regardless of** computer resources necessary for the low-priority process to run. Claim 16 is thus patentable at least for any of the reasons advanced with respect to claim 1 and claim 14.

Claim 16 is also patentable on its own merits since claim 16 recites other features of the invention neither disclosed, taught nor suggested by the applied art. More particularly, claim 16 is directed to a resource reallocation process in a specific situation when a node, on

which a high-priority process is running, fails. As discussed in the foregoing sections, especially at point (h), the applied art of record, especially Culbert, handles this situation differently from the claimed method.

Conclusion

For the reasons advanced above, Appellant respectfully requests that the *35 U.S.C. 103(a)* rejection of claims 1-5 and 9-16 be reversed.

B. Second Issue

35 U.S.C. 103(a) rejection of claim 7 as being unpatentable over Culbert in view of Sumimoto and further in view of Hauser

The Examiner rejected claim 7 under *35 U.S.C. 103(a)* as being unpatentable over Culbert in view of Sumimoto and further in view of Hauser. Appellant respectfully traverses this rejection at least for the reason advanced with respect to claim 1.

Conclusion

Appellant respectfully requests that the *35 U.S.C. 103(a)* rejections of claim 7 be reversed.

Each of the Examiner's rejections has been traversed. Accordingly, Appellant respectfully submits that all claims on appeal are considered allowable. Reversal of the Examiner's Final Rejection is believed appropriate and courteously solicited.

If for any reason this Appeal Brief is found to be incomplete, or if at any time it appears that a telephone conference with counsel would help advance prosecution, please telephone the undersigned, Appellant's attorney of record.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 07-1337 and please credit any excess fees to such deposit account.

Respectfully submitted,

LOWE HAUPTMAN GILMAN & BERNER, LLP



Kenneth M. Berner
Registration No. 37,093

USPTO Customer No. 22429
1700 Diagonal Road, Suite 310
Alexandria, Virginia 22314
(703) 684-1111
(703) 518-5499
Date: April 29, 2002
BJH:lcw

IX. APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

1. A method of allocating network resources on a computer network, comprising:
monitoring at least two nodes on the computer network among at least a first process and a second process for allocation of computer resources on each of the at least two nodes;
assigning a priority to each of the at least two processes, the second process being assigned a lower priority than the first process;
for the first process running on at least one of the two nodes, setting a minimum resource allocation for the first process on the at least two nodes independent of the computer resources needed by other processes running on the computer network; and
redistributing computer resources on the network so that the minimum resource allocation for the first process is guaranteed should insufficient network resources be available.
2. The method of claim 1, wherein the minimum resource allocation is an allocation of computers and memory space on the network for the first process.
3. The method of claim 1, wherein the minimum resource allocation is an allocation of a minimum percentage of resources at the at least two nodes for the first process and for the second process.
4. The method of claim 1, wherein said monitoring step is performed periodically.
5. The method of claim 1, wherein said monitoring step is performed continually and the minimum resource allocation is an allocation of a minimum percentage of resources at the at least two nodes for the first process and for the second process.
7. The method of claim 1, comprising the further step of setting a maximum resource allocation for at least one of the first process and the second process.

9. The method of claim 1, comprising the further step of storing the minimum resource allocation in a storage device.

10. The method of claim 1, wherein said monitoring step can be performed by any of the at least two nodes on the computer network.

11. An article, comprising:
at least one sequence of machine executable instructions in machine readable form,
wherein execution of the instructions by one or more processors causes the one or more processors to:

- (i) monitor at least two nodes on the computer network among at least two processes for allocation of computer resources on each of the at least two nodes;
- (ii) assigning a priority to each of the at least two processes, the second process being assigned a lower priority than the first process;
- (iii) for a first process of the at least two processes running on at least one of the two nodes, set a minimum resource allocation for the first process on the at least two nodes irrespective of the computer resources needed by other processes running on the computer network; and
- (iv) redistributing computer resources on the network so that the minimum resource allocation for the first process is guaranteed should insufficient network resources be available.

12. A computer architecture for switching resource allocation policies on a computer network, comprising:

monitoring means for monitoring at least two nodes on the computer network among at least a first and a second process for allocation of computer resources on each of the at least two nodes;

assigning means for assigning a priority to each of the at least two processes, the second process being assigned a lower priority than the first process;

for the first process running on at least one of the two nodes, setting means for setting a minimum resource allocation for the first process on the at least two nodes independent of the computer resources needed by other processes running on the computer network; and

redistributing means for redistributing computer resources on the network so that the minimum resource allocation for the first process is guaranteed should insufficient network resources be available.

13. A computer system comprising:

a processor; and

a memory coupled to said processor, the memory having stored therein sequences of instructions, which, when executed by said processor, cause said processor to perform the steps of:

monitoring at least two nodes on the computer network among at least a first process and a second process for allocation of computer resources on each of the at least two nodes;

assigning a priority to each of the at least two processes, the second process being assigned a lower priority than the first process;

for the first process running on at least one of the two nodes, setting a minimum resource allocation for the first process on the at least two nodes independent of the computer resources needed by other processes running on the computer network; and

redistributing computer resources on the network so that the minimum resource allocation for the first process is guaranteed should insufficient network resources be available.

14. The method of claim 1, wherein said redistributing step is performed by removing a computer resource previously assigned to the second process, and reallocating the removed computer resource to the first process irrespective of an amount of computer resources necessary for the second process to run on the computer network.

15. The method of claim 1, wherein said assigning step is performed irrespective of amounts of computer resources necessary for each of the at least two processes to run on the computer network.

16. A method of allocating network resources on a computer network, comprising the steps of:

allocating computer resources on each node of the computer network among at least a high-priority process and a low-priority process according to an allocation schema associated with said node;

setting a minimum resource allocation for the high-priority process independent of computer resources needed by other processes running on the computer network;

monitoring first nodes on which the high-priority process is executed; and

if computer resources allocated to the high-priority process on at least one of said first nodes become unavailable, changing the allocation schema associated with at least one of second nodes on which the low-priority process is executed to remove an amount of computer resources previously assigned to the low-priority process and reallocate the removed amount of computer resources to the high-priority process, whereby the minimum resource allocation for the high-priority process is guaranteed regardless of computer resources necessary for the low-priority process to run on said at least one of said second nodes.